

**Method for accessing an object using a 'WEB' brow co-operating with a smart card and support therefor**

Publication number: CN1300494 (A)

Publication date: 2001-06-20

Inventor(s): URIEN PASCAL [FR]

Applicant(s): BULL CP8 CO [FR]

Classification:

- International: G06K19/00; G06F12/00; G06K17/00; G06K19/06;  
H04L29/06; H04L29/08; G06K19/00; G06F12/00; G06K17/00;  
G06K19/07; H04L29/06; H04L29/08; (IPC1-7): H04L29/06

- European: H04L29/06; H04L29/08N1; H04L29/08N27R

Application number: CN20008000590 20000315

Priority number(s): FR19990003172 19990315

## Also published as:

CN100385889 (C)

FR2791159 (A1)

US6944650 (B1)

JP2006134335 (A)

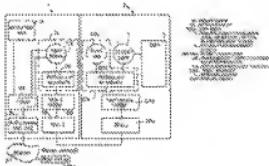
JP2002539546 (T)

more &gt;

Abstract not available for CN 1300494 (A)

Abstract of corresponding document: FR 2791159 (A1)

The invention concerns an architecture for securely accessing virtual objects (Obv???) distributed in systems connected to the Internet (R), and for obtaining an instance thereof. The access is performed via a smart card (2a), through a "WEB" browser (10). The terminal (1) and the smart card (2a) comprise each a specific protocol layer (13, 23a). The latter contains intelligent agents (13, 23a1) for setting up two-way data exchange sessions, thereby enabling the smart card (2a) to have a "WEB" server functionality. The smart card (2a) also comprises intelligent agents, called script translators, and a virtual file management system (8) co-operating with a specialised script translator intelligent agent (7). Each virtual object (Obv???) is associated with a virtual file of the virtual file management system (8). The specialised intelligent agent (7) presents to the browser (10) the list of accessible virtual objects (Obv???) and generates methods for accessing said objects.



Data supplied from the esp@cenet database — Worldwide

## [12] 发明专利申请公开说明书

[21] 申请号 00800590.7

[43] 公开日 2001 年 6 月 20 日

[11] 公开号 CN 1300494A

[22] 申请日 2000.3.15 [21] 申请号 00800590.7  
 [30] 优先权  
 [32] 1999.3.15 [33] FR [31] 99/03172  
 [86] 国际申请 PCT/FR00/00625 2000.3.15  
 [87] 国际公布 WO00/56030 法 2000.9.21  
 [85] 进入国家阶段日期 2000.12.14  
 [71] 申请人 布尔 CP8 公司  
 地址 法国卢旺茨那斯  
 [72] 发明人 帕斯卡尔·乌林

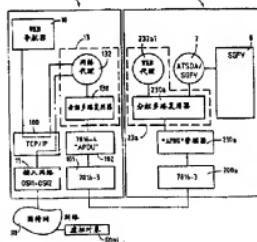
[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所  
 代理人 马 浩

权利要求书 3 页 说明书 30 页 附图页数 8 页

[54] 发明名称 借助于“WEB”导航器与一个灵巧卡协作存取对象的方法和用于应用该方法的架构

## [57] 摘要

发明涉及到一种方法和架构，它能够以安全的形式存取分布在与互连网络(RI)相连接的系统中的虚拟对象(Obv<sub>i</sub>)，并且获得一个示例。该存取通过一灵巧卡(2a)，借助于一个“WEB”型导航器(10)来被执行。终端(1)和灵巧卡(2a)的每一个包含一个特殊的协议层(13, 23a)。该后者包括用于建立数据双向交换会话的智能代理(132, 232a)，它能使灵巧卡(2a)具有一个“WEB”服务器的功能。灵巧卡(2a)也包括称为脚本翻译器的智能代理，以及与一个特殊的脚本翻译器智能代理(7)协同工作的虚拟文件的管理系统(8)。每个虚拟对象(Obv<sub>i</sub>)与虚拟文件管理系统的一个虚拟文件相关联，特殊的智能代理(7)向导航器(10)展示可存取的虚拟对象(Obv<sub>i</sub>)列表，并管理存取这些对象的方法。



## 权 利 要 求 书

1. 一种嵌入系统，它配备有一个包括信息处理装置和信息存储装置，并用于通过终端与网络协同工作的灵巧卡，其特征在于：
  - 它存储至少一个对象文件，它容纳有与网络上一对对象相关联的信息并且能够实现该对象的例示；
  - 它包括用于与位于终端中的相匹配的网络接口部件协同工作的网络接口装置，使得嵌入系统构成网络上的一个信息服务器；以及
  - 它包括对象文件接口装置，用于在网络接口装置所传送并分配到至少所述对象文件的信息和与所述对象文件交换的信息之间建立对应。
2. 根据权利要求 1 所述的嵌入系统，在其中对象文件包括一个在导航器软件上执行的自主软件片。
3. 根据权利要求 1 所述的嵌入系统，在其中所述的网络接口装置被安置得以与位于终端中的相匹配的网络接口装置协同工作，以使得嵌入系统如同一个客户一样能够与至少一个网络服务器相连接。
4. 一种用于例示一位于一网络上的对象的方法，其特征在于它使用嵌入系统，系统配备有一个包括信息处理装置和信息存储装置，并用于通过终端与网络协同工作的灵巧卡，嵌入系统存储至少一个对象文件，它容纳有与网络上一对对象相关联的信息并且能够实现该对象的例示，并且一方面包括用于与位于终端中的相匹配的网络接口部件协同工作的网络接口装置，使得嵌入系统构成网络上的一个信息服务器；以及另一方面包括对象文件接口装置，用于在网络接口装置所传送并分配到至少所述对象文件的信息和与所述对象文件交换的信息之间建立对应，方法的特征还在于它借助于至少下列步骤通过一个对象文件来描述一个代理之间的会话集：
  - 建立一个应用代理表；
  - 对于每一个代理，定义代理所必需的调用参数。
5. 根据权利要求 4 所述的方法，在其中一个调用参数描述了与另

一个代理会话的发起。

6. 根据权利要求 4 所述的方法，其中一代理修改另一代理所使用的列表。

7. 一种用于访问一位于一网络上的对象的方法，其特征在于它使用嵌入系统，系统配备有一个包括信息处理装置和信息存储装置并用于通过终端与网络协同工作的灵巧卡，嵌入系统存储至少一个对象文件，它容纳有与网络上一对象相关联的信息并且能够实现该对象的例示，并且一方面包括用于与位于终端中的相应的网络接口部件协同工作的网络接口装置，使得嵌入系统构成网络上的一个信息服务器；以及另一方面包括对象文件接口装置，用于在网络接口装置所传送并分配到至少所述对象文件的信息和与所述对象文件交换的信息之间建立对应，方法特征还在于它通过至少下列步骤实现嵌入系统信息服务器执行的对象文件所描述的代理之间的会话：

- 标识对象文件；
- 执行该对象文件。

8. 根据权利要求 7 所述的方法，其中通过例示与对象文件相关联的第一代理。

9. 根据权利要求 7 所述的方法，其中通过例示一个或多个对象文件所参考的第一代理来执行。

10. 一种用于例示一位于一网络上的对象的方法，其特征在于它使用嵌入系统，系统配备有一个包括信息处理装置和信息存储装置，并用于通过终端与网络协同工作的灵巧卡，嵌入系统存储至少一个对象文件，它容纳有与网络上一对象相关联的信息并且能够实现该对象的例示，并且一方面包括用于与位于终端中的相应的网络接口部件协同工作的网络接口装置，使得嵌入系统构成网络上的一个信息服务器；以及另一方面包括对象文件接口装置，用于在网络接口装置所传送并分配到至少所述对象文件的信息和与所述对象文件交换的信息之间建立对应，方法另外的特征在于它通过至少下列步骤实现导航器软件执行的对象文件所描述的代理之间的会话：

- 由一对象文件的导航软件加载并且一个特殊的软件能够实现的软件；

- 由导航器软件执行该特殊的软件。

11. 根据权利要求 10 所述的方法，在其中特殊软件通过由导航器软件可执行的各种解译语言来实现。

12. 根据权利要求 10 所述的方法，在其中对象文件解释器在导航器软件上被实现。

13. 一种用于访问一位于一网络上的对象的方法，其特征在于它使用嵌入系统，系统配备有一个包括信息处理装置和信息存储装置并用于通过终端与网络协同工作的灵巧卡，嵌入系统存储至少一个对象文件，它容纳有与网络上一对象相关联的信息并且能够实现该对象的例示，并且一方面包括用于与位于终端中的相匹配的网络接口部件协同工作的网络接口装置，使得嵌入系统构成网络上的一个信息服务器；以及另一方面包括对象文件接口装置，用于在网络接口装置所传送并分配给至少所述对象文件的信息和与所述对象文件交换的信息之间建立对应，方法另外的特征在于它能够使得嵌入系统实现在由导航器软件执行的一对对象文件所描述的代理之间的会话，并且它包括通过统一资源标识符来识别用于实现导航器软件的特殊的软件的步骤。

14. 根据权利要求 13 所述的方法，其中统一资源标识符被集成在一封文文档中。

15. 根据权利要求 13 所述的方法，在其中所述特殊的软件由一在导航器软件上可用的方法所装载并且从统一资源标识符推导出。

## 说 明 书

借助于“WEB”导航器与一个灵巧卡协作  
存取对象的方法和用于应用该方法的架构

发明涉及到一种容纳有能够例示位于网络中的对象的信息的嵌入系统，以及一个用于例示该对象的方法。

更特别地，发明还涉及到一种安全存取该对象的方法。

在发明的范围内，术语“对象”必须从其最一般的的意思来考虑。它包括多种信息源，例如文本文件，图像文件或多媒体文件（图像，声音，等等）。它还包括基于一给定协议的信息系统的交易和链接。

在第一种情况中，我们将在后面称其为静态对象，因为它们的请求不依赖于时间。在第二种情况下中，我们将称其为动态对象，因为它们的访问根据时间是可变的。作为一个非限制性的例子，在因特网类型的网络范围内，我们引入一个“Telnet”连接。

同样在发明的范围内，术语“用户站”必须是被包括在一般的含义之中。前述的用户站特别是可以由一台在不同操作系统，例如WINDOWS或UNIX（所有二者都是注册商标）下工作的个人计算机。它同样可以由一台工作站，一台便携计算机或一个称为“专用”的卡终端。

同样，在发明的范围内，术语“网络”包括所有的带有一组相互连接的服务器的网络，特别是一个全球网络，在其中信息被端到端地传输。这特别是涉及到因特网，在网络中数据交换根据因特网类型，公司的专用网或类似的，称为“intranet（内部网）”，以及朝向外部延伸的网络，被称为“extranet（外延网）”的协议进行。这还特别涉及到如同例如I3E802.11，BLUE TOOTH的被称为“Wireless Network（无线网络）”的一个GSM（英文Global System Mobile（全球移动系统）），ATM，UMTS，GPRS（英文Global Packet Radio System（全球分组无线系统））网络。

接下来，并不限制范围，我们将处于发明的一个最佳的应用中，除非有其它的说明。因此我们将考虑一个用户站，我们将简单地称其为“终端”，它带有一个灵巧卡阅读器，并且与一个因特网类型的网络连接。

一基于灵巧卡的应用系统通常包括下列主要部件：

- 一个灵巧卡；
- 一个构成前述终端的主机系统；
- 一个通讯网络，即在最佳应用中的因特网网络；
- 以及一个与网络连接的应用服务器。

图 1 简要提示出了该类型的一个架构的例子。终端 1，例如是一个独立的计算机，包括一灵巧卡 2 的阅读器 3。该阅读器 3 可以是也可以不是物理性的集成在终端 1 中。灵巧卡 2 包括一个集成电路 20，其输入-输入的连接位于其支撑表面上，以便能够供应电能及与终端 1 通讯。该终端包括接入数据传输网络 RI 的电路。这些电路特别是依赖于 RI 网络和终端 1 的特性。作为例子，这可能涉及到对于局域类型网络的一块网卡，或对于连接到电话交换线路或综合服务数字网（“RNIS”）的一个调制解调器，以便接入到因特网网络，例如经过一个因特网服务提供商（英文术语是“Internet Service Provider”或“ISP”）。

自然地，终端 1 包括所有使其良好运行所必需的电路和构件，并且为了减化绘图的目的而未被示出：中央单元，随机存取存储器和只读存储器，磁盘大容量存储器，软盘驱动器和/或 CD-ROM，等等。

通常，终端 1 还连接到一些传统的，集成的或非集成的外围设备，例如一个可视化的屏幕 5 和一个键盘 6。

终端 1 可以与服务器或所有与 RI 网络连接的信息系统通讯，其中唯一的一个，4，被标识在图 1A 上。对于“服务器”，我们可以理解为所有用于处理通讯协议的信息服务器，或者是给与存取文档，或者是给与存取机器。在发明最佳应用的情况下，接入电路 11 利用被称“WEB”类型的导航器，或英文“browser（浏览器）”的特殊软

件 10 使得终端 1 与服务器 4 通讯。这就允许接入分布在整个 RI 网络上的不同应用，一般是根据“客户-服务器”的模式来进行的。对于“导航器”，我们理解为提供了以下功能的所有的设备：

- 显示一页，特别是一个“SGML”（英文“Standard Generalized Markup Language”或标准通用标记语言）标准的页；
- 页中所提供的资源的下载。

该导航器的功能对应于英文术语“browser”所具有的功能。一个 SGML 页显示属性，或链接到其它 SGML 文档，或朝向外部世界的“超链接”，也就是说 URI（英文 Unified Resource Identifier，统一资源标识符）。

关于 SGML 语言，公知地包括多种分支，HTML，XML 和 WML。

习惯上，在网络上的通讯依据符合包括多个叠置软件层标准的协议来进行。在因特网类型的 RI 网络的情况下，通讯根据特定于该通讯类型的协议来进行，这将在后面详细描写，但它同样包括多个软件层。通讯协议根据所特别面对的应用来被选择：“WEB”页询问，文件传输，电子邮件（电子邮件或英文“e-mail”），论坛或新闻（英文 news），等等。

包括一个终端，一个灵巧卡阅读器和灵巧卡的系统的逻辑架构，由图 1 所简单地表示。它由标准 ISO 7816 所描述，它本身包括多个子组：

- ISO 7816-1 和 ISO 7816-2，涉及到卡的尺寸和标记；
- ISO 7816-3，涉及到终端和灵巧卡之间的数据传输，并且
- ISO 7816-4，涉及到指令组的结构和命令格式。

在图 1B 上，在终端 1 这边，我们只示出了符合标准 ISO 7816-3，标记为 101 的层，以及“APDU”指令处理器（gestionnaire）（ISO 7816-4），标记 102。在灵巧卡 2 这边，符合标准 ISO 7816-3 的层被标记为 200，而“ADPU”（标准 7816-4）指令处理器被标记为 201。应用被标记为  $A_p, \dots, A_p, \dots, A_n$ ； $n$  是在灵巧卡 2 上所存

在的最大应用数。

存在于灵巧卡 2 (图 1A) 中的一个“cartlet (小卡)” (注册商标) 应用通过一个指令组与终端 1 对话。该套指令典型的具有写指令和读指令。指令的格式以英文缩写“APDU” (指“Application Protocol Data Unit (应用协议数据单元)”) 而是公知的。它由前述的标准 ISO 7816-4 所定义。一命令的“APDU”被记为“APDU.command”，并且一响应的“APDU”被记为“APDU.response”。“APDU”在卡的阅读器和灵巧卡之间通过前述标准 ISO 7816-3 所指定的协议 (例如以字符模式: T=0; 或以块模式: T=1) 来被交换。

当灵巧卡 2 包括许多不同的应用时, 如同我们在图 1B 上所示的那样, 我们称为多应用卡。然而, 终端 1 同时只与一个应用对话。一个应用  $A_i$  具有例如一个以 JAVA (注册商标) 语言编写的软件, 被称为“applet (小应用)”, 我们以后将其称为“cardlet”。一特殊“cardlet” $A_i$  的选择通过选择类型 (“SELECT”) 的“APDU”来被获得。当该选择被执行后, 接着的“APDU”就被朝向该“cardlet”发送。一个新的“APDU SELECT”将会放弃正在执行的应用并选择另一个。“APDU”管理软件子集合 201 能够在灵巧卡 2 中选择一个特殊的应用  $A_i$ , 存储如此选择的应用, 并且朝向该应用传输或从该应用接收“APDU”。

作为刚才所描述的一个简述, 一应用  $A_i$  的选择及与其的对话通过交换“APDU”指令来进行。我们假设应用  $A_i$  是常规应用, 我们此后将其称为“CGA” (指“Generic Card Application”或通用卡应用)。

在基于灵巧卡的一个应用系统中, 如同图 1B 架构所示的那样, 灵巧卡可以被看作具有不同的功能, 特别是安全功能。实际上其优点是在由用户保存的灵巧卡中存储与安全相关的数据 (口令, 存取权限, 等等)。另外, 被记录在只读存储器中的数据, 它可以以加密的形式, 不能容易地被修改, 也不能被外部直接读取。

然而，要注意到卡 3 不能与商用导航器直接通讯，除非修改该导航器的编码。实际上的灵巧卡，另外它还符合前面所提到的标准，具有一个更加不能直接与因特网网络通讯的硬件和软件配置。特别地，它不能根据在该类型网络上所使用的这个或那个协议接收或发送数据包。因此这就需要一个置入终端 1 中的附加软件片，它通常具有一个根据英文术语被称为“*plug-in (插件)*”的形式。该软件片段，在图 1 上带有标记 12，在导航器 10 和卡 2，更准确地说是该卡 2 的电子电路 20 之间执行一个接口作用。

在实际的技术领域中，与卡阅读器 3 相关的主机系统，也就是说终端 1，还与一个特殊的应用相关联。换句话说，对于每一个特殊的应用，必须有一个特定终端，称为“专用”。

另外，尽管考虑到技术的快速发展以及将来可以预见的变化，很清楚的是灵巧卡存储电路中，随机地或只读的，记录信息的容量是并且会是非常有限的，如果我们把这种容量与该灵巧卡的“主机”终端相比较，以及与更加大型的系统，“微型计算机”或被称为“大型机”类型的大系统相比较。因此，不可能在灵巧卡中存储大量应用的数据，特别是体积庞大的多媒体文件。

发明在于克服现有技术装置的缺陷，一些现有技术的装置已经介绍过了，并都符合所显示出来的需求。特别是必须要能够接入大量的应用，虽然从数据量这一点看是大量的，不同性质并且分布在整個因特网网络上。另外，在一个最佳实施例中，存取必须享有一个最大的安全性，也就是说实践中通过或在容纳有所有数据交换安全性所必需的数据的灵巧卡的控制下进行。最后，该存取必须能够由商业导航器来实现和对于一个用户来说是透明的用户只能通过灵巧卡作为“源”“看到”，而不论应用存储在哪里。

根据方法的一个特性，灵巧卡具有一个“主机”系统，也就是说终端，一个虚拟终端模块，例如具有“HTML”（指“HyperText Markup Language（超文本标记语言）”）语言的形式，或更加一般的超文本语言，或还可以是 JAVA（注册商标）语言的“applet”，它

可以允许用户在可用的并且是灵巧卡所拥用的应用中选择特定的应用。因此，终端是普通的并且支持多个应用。主机系统被看作是灵巧卡的一个外围设备，并且在其配置上具有硬件资源，例如一个可视化屏幕，一个键盘，等等。

为此，我们在灵巧卡以及其在终端的对应部件之间有一个专用的通讯软件层。术语“特殊的”必须被理解为对于发明方法为专用的。实际上，这些被称为特殊的通讯层是通用的，而不论所考虑的应用。它只在一方面灵巧卡和终端之间，另一方面在灵巧卡和网络之间的双向数据交换处理中介入。

特殊的通讯软件层特别是包括了被称为“智能代理”的软件部件，它能进行特别的协议转换。分别在与终端和灵巧卡相关联的特定的通讯层中存在有所配备的智能代理。根据发明的方法，在所配备的智能代理之间建立会话。

根据第二个特性，发明方法提供了一种激活位于灵巧卡中的常规类型应用的可能，也就是前面提到的“CGA”类型，而不论它是什么而不必进行修改。

为此我们配备一个或多个被称为脚本翻译器的智能代理，它接收导航器的请求并将它们翻译成“CGA”类型应用能够理解的“APDU”指令。该技术特性使其能够被置入到灵巧卡中，其架构符合发明的方法，一个被置入到传统的“WEB”服务器中的与被称为“CGI”（指“Common Gateway Interface（通用网关接口）”）功能类似的机制。

最后，根据发明方法的另一个特征，通过将前述功能和机制应用，这就可以允许访问分布在连接有终端的数据传输网络上的信息资源，特别是因特网网络或等效的网络（内部网，外部网），而对于用户不必考虑他所处的位置。在后面，如同已经指出的那样，这些资源将被称为“虚拟对象”，静态的或动态的。

为此，将使用用于该任务的脚本翻译器智能代理并与终端和/或灵巧卡中存在的其它智能代理协同工作。该代理一方面能够定义灵巧卡

以及同样对于用户（或灵巧卡的持有者）能够访问的虚拟对象，并且另一方面通过灵巧卡向询问导航器提供了能够访问这些虚拟对象的方法。

因此发明涉及了一种嵌入系统，它配备有一个包括信息处理装置和信息存储装置，并用于通过终端与网络协同工作的灵巧卡，其特征在于：

- 它存储至少一个对象文件，它容纳有与网络上一对象相关联的信息并且能够生成该对象的示例；
- 它包括用于与位于终端中的相应的网络接口部件协同工作的网络接口装置，使得嵌入系统构成网络上的一个信息服务器；以及
- 它包括对象文件接口装置，用于在网络接口装置所传送并由至少所述对象文件所执行的信息与所述对象文件交换的信息之间建立对应。

更好地，对象文件包括一个在导航器软件上执行的自主软件片。更好地，该自主软件片使得嵌入系统的对象文件管理系统运行。

更好地，所述的对象文件包括用于例示一对象而实现的动作的描述。更好地，动作包括嵌入系统内部实现的动作并且由嵌入系统代理之间的会话所构成。

更好地，动作包括在嵌入系统外部的动作并且由与终端代理的会话所构成，以便从网络信息服务器获得信息。

更好地，所述的网络接口装置被安置得以与位于终端中的、对应的网络接口装置协同工作，以使得嵌入系统如同一个客户一样包括与至少一个网络服务器相连接的能力。

因此，发明涉及一种使用前述嵌入系统并用于例示位于网络上对象的方法，其特征在于它借助于至少下列步骤通过一个对象文件来描述一个代理之间的会话集：

- 建立一个应用代理表；
  - 对于每一个代理，定义代理所必需的调用参数。
- 更好地，一个调用参数描述了与另一个代理会话的发起。

更好地，一代理修改另一代理所使用的变元表。

作为变化，方法的特征在于它通过至少下列步骤实现嵌入系统信息服务器执行的对象文件所描述的代理之间的会话：

- 对象文件的标识；
- 该对象文件的执行。

更好地，该对象文件由一个特殊的目录名来标识。

更好地，该对象文件由一个特殊的文件属性来标识。

更好地，该对象文件由一个特殊的命名准则来标识。

更好地，该对象文件由与对象文件相关联的第一代理的示例来执行。

更好地，该对象文件由一个或多个由对象文件参照的代理的示例来执行。

作为变化，方法的特征在于它通过至少下列步骤实现导航软件执行的对象文件所描述的代理之间的会话：

- 由一对象文件的导航软件所加载和一能够使其应用的软件；
- 由导航软件执行特殊的软件。

更好地，特殊软件通过由导航软件可执行的、各种解译语言来实现。

更好地，对象文件解释器在导航软件上被实现。

作为变化，方法的特征是它能够使得嵌入系统提供在由导航器软件执行的一对象文件所描述的代理之间实现会话的可能，并且它包括构成通过统一资源标识符来识别特殊的、用于使导航器软件工作的软件的步骤。

更好地，统一资源标识符被集成在一超文本文档中。

更好地，所述的超文本文档被容纳在嵌入系统中。

更好地，所述的超文本文档被容纳在远离嵌入系统的一个网络信息服务器上。

更好地，所述特殊的软件由一在导航软件上可用的方法所装载并且从统一资源标识符导出。

发明还涉及一种嵌入系统，它带有一个灵巧卡，灵巧卡包括信息处理装置和信息存储装置，并且用于通过一终端与一网络协同工作，其特征在于它包括网络接口装置，被安置的用于与位于终端中的、相匹配的网络接口装置协同工作，以便嵌入系统在网络上构成一个信息服务器，和/或如同一个客户一样能够连接至少一个网络服务器。发明还涉及一个用于与一网络协同工作的终端，并且包括信息处理装置，信息存储装置和与一嵌入系统协同工作的装置，它带有一包括信息处理装置和信息存储装置的灵巧卡，其特征在于它包括网络接口装置，被安置的用于与位于嵌入系统中的、相匹配的网络接口装置协同工作，以便嵌入系统在网络上构成一个信息服务器，和/或如同一个客户一样能够连接至少一个网络服务器。更好地，终端由一软件装载机制从网络中动态地获得所述的网络接口装置。这特别的涉及到英文术语插入（“plug-in”）的机制。如果网络不知道容纳有所述软件的文件的扩展，这将在服务器上查找与该扩展相关的软件，这通常被称为“帮助（helper）”。

更好地，所述相匹配的网络接口装置在终端和在嵌入系统中构成一个栈，它包括一个或多个通讯层，例如它允许嵌入系统共享终端通讯层的所有或部分。另外，终端最好具有在这些通讯层上的接入点，这允许它从或朝向这些层的一个或多个路由信息流。这些接入点对应于由 ISO 标准所定义以“SAP”（英文 Service Access Point（服务接入点））名称所公知的点。

发明还涉及一种嵌入系统，它带有一个灵巧卡，包括信息处理装置和信息存储装置，并且用于通过一终端与一网络协同工作，其特征在于它包括网络接口装置，被安置的用于与位于终端中的、相匹配的网络接口装置协同工作，以便至少一部分在终端应用和网络之间交换的信息流根据终端公知的标准通过嵌入系统的网络接口装置。发明还涉及一个用于与一网络协同工作并且包括信息处理装置，信息存储装置和与一嵌入系统协同工作的装置的终端，它带有一包括信息处理装置和信息存储装置的灵巧卡，其特征在于它包括网络接口装置，被安

置的用于与位于嵌入系统中的、相匹配的网络接口装置协同工作，以便至少一部分在终端应用和网络之间交换的信息流根据终端公知的标准通过嵌入系统的网络接口装置。朝向嵌入系统的一部分信息流的路由最好由终端信息处理装置根据预先建立的标准来实现，或者以静态的方式，或者通过与嵌入系统对话的协商方式；在后者情况下，终端可以例如向嵌入系统通过公知的协议请求“IP”地址（如果这涉及因特网网络）。更好地，前述的标准包括下列之一：

嵌入系统的“IP”地址，或在 ATM 网络情况下的其 ATM 地址；

终端的 IP 地址以及一特殊的“TCP”端口或“UDP”端口；

所有参考于嵌入系统的 SAP 接入点。

发明将通过对附图的详细描述而来被描述，在其中：

- 图 1A 和 1B 分别简要地示出了根据现有技术基于灵巧卡的应用系统的一个例子的硬件和软件架构；

- 图 2 简要地示出了根据发明的基于灵巧卡的应用系统的一个例子，后者还涉及到“WEB”服务器；

- 图 3 以简要的逻辑架构的形式示出了灵巧卡包括智能代理的一个系统；

- 图 4 示出了一个符合发明的系统架构，其中灵巧卡包括脚本翻译器智能代理；

- 图 5 是一个简要地示出了在一个导航器和一个具有图 4 架构的灵巧卡之间交换阶段的框图；

- 图 6 是一个简要地示出根据发明法的一个基本要点的框图，通过它可以通过一灵巧卡和一“WEB”类型导航器访问分布在因特网类型网络上的虚拟对象；

- 图 7 简要地示出了为使该发明方法要点能够应用的被称为虚拟的文件管理系统的构造；

- 图 8 是一个包括根据图 7 的虚拟文件管理系统的架构例子；

- 图 9 到 15 示简要地示出了根据发明的方法的多个实施例的框

图。

在描述位于一根据发明的灵巧卡的应用激活方法以及详细描述使其应用的架构之前，似乎首先简要地回顾一下网络上通讯协议的原理特性是有用的。

通讯网络的架构有不同的层来被描述。做为例子，由 ISO 定义的标准“OSI”（“Open System Interconnection（开放系统互连）”）包括 7 层，它从被称为底层的层（例如涉及到物理传输支持的被称为“物理”的层）到被称为高层的层（例如被称为“应用”的层），通过中间层，特别是被称为“传输”的层。一个给定的层通过适当的接口向其紧邻的上层提供其服务，并且从其紧邻的下面的层请求其它的服务，通过合适的接口。层通过原语通讯。它同样可以与同级的层通讯。在某些架构中，这些层的这个或那个可能存在。

在因特网类型的环境中，层是 5 个，更准确地说，从上面的层到下面的层：应用层（“http”，“ftp”，“e-mail”，等等），传输层（“TCP”），网络地址层（“IP”），数据链接层（“PPP”，“Slip”，等等）和物理层。

进行完这些回顾后，我们现在开始描述一个基于灵巧卡的应用系统的架构，它能够使得灵巧卡工作得如同一个“WEB”服务器。图 2 简要地示出了一个该架构的例子。图 1A 和图 1B 上相同的部件具有相同的标号并且仅是在需要的时候才被重复描述。为简化绘图，我们没有示出与终端相连的不同外围设备（图 1A：例如屏幕 5 和键盘 6）。

除了分别被置于终端 1 和灵巧卡 2a 中的特定通讯协议软件层外，标记为 13 和 23a，其它的部件，硬件或软件与现有技术的相同。

终端 1 包括 RI 网络接入电路 11，它由例如对于因特网是一个调制解调器，对于局域网是一个网卡组成。这些电路集中对应于“物理”和“数据链接”层的下面软件层 C1 和 C2。

我们同样表示对应于“网络地址”（在因特网情况下是“IP”）和“传输”（“TCP”）层的上面层 C3 和 C4。应用的上面层（“http”，“ftp”，“e-mail”，等等）没有被表示。

下面层 C1 和 C2 以及上面层 C3 和 C4 之间的接口通常由被称为“底层驱动”软件层构成。上面层 C3 和 C4 依靠该接口上并且通过它们所对应的特殊函数库架或网络程序库 14 而被使用。在因特网网络的情况下，“TCP/IP”通过被称为“网络程序接口（sockets）”的程序库而被使用。

该构造允许一导航器 10（图 1A）向一服务器 4（图 1A）提交请求，以查询“WEB”页（“HTTP”协议），传输文件（“FTP”协议）或发送电子邮件（“e-mail”协议），这些都是传统的方式。

终端 1 同样包括一个集成的或非集成的卡阅读器 3。为与灵巧卡 2a 通讯，卡阅读器同样包括两个底层 CC1（物理层）和 CC2（数据链接层），它们与 C1 和 C2 层起相同的作用。与层 CC1 和 CC2 的软件接口由例如规范“PC/SC”（“第六部分，服务提供商”）所描述。CC1 和 CC2 层本身特别是由如同前面已经回顾过的标准 ISO 7816-1 到 7816-4 所描述。

辅助软件层 16 在应用层（未示出）和下面层 CC1 和 CC2 之间形成接口。对于该层的主要功能是多路传输/多路分解功能。

与灵巧卡 2a 的通讯根据一个类似于在“UNIX”（注册商标）类型操作系统中文件操作所使用的范例来被执行：打开（“OPEN”），读（“READ”），写（“WRITE”），关闭（“CLOSE”），等等。

从灵巧卡 2a 这边，我们发现一个类似的结构，即存在两个基层，标记为 CCa1（物理层）和 CCa2（数据链接层），以及一个完全类似于层 16 的接口层 26a。

根据第一个特性，我们在终端 1 和在灵巧卡 2a 中设置两个特殊的协议层，分别为 13 和 23。

在终端 1 中，特定层 13 与“底层驱动”15，与网络层程序库 14，C3 和 C4，以及卡阅读器 3 的协议层，也就是说下面层，CC1 和 CC2，通过多路传输层 16 接口。特定层 13 能够从和朝向灵巧卡 2a 传送网络分组。另外，对于灵巧卡 2a 的应用，它适于现有的应用，例如

因特网导航器（图 2），电子邮件，等等。

从灵巧卡 2a 这边，我们发现一个与层 13 对应的、标记为 23a 的特定层的辅助例示构成的完全类似的机构。

更准确地，特定层，13 和 23a，被划分为 3 个主要软件部件：

- 一个在层 13 和 23a 之间经过常规层 CC1, CC2, CCa1 和 CCa2 的信息块传输模块 130 或 230a;
- 一个或多个被称为“智能代理”的软件片，132，或 232a，它实现，例如，协议转换的功能；
- 以及一个特殊配置管理模块，分别为 131 和 231a，模块可被看作是一个特殊的智能代理。

因此在终端 1 和灵巧卡 2a 中找到一个两实体间的通讯协议栈。

第二级的层 CC2 和 CCa2（数据链层）保证了在灵巧卡 2a 和终端 1 之间的交换。这些层用于传输检错和纠错。可以使用不同的协议，并且作为非限制性的例子如下：

- 建议 ETSI GSM 11.11;
- 由标准 ISO 7816-3 定义的协议，以字符模式 T=0;
- 由标准 ISO 7816-3 定义的协议，以块模式 T=1;
- 或由标准 ISO 3309，以“HDLC”帧模式（指“High-Level Data Link Control procedure”，高级数据链接控制规程）。

在发明的范围内，我们最好使用 ISO 7816-3 协议，以块模式。

以本身公知的方式，对于每一个协议层，关联一些数目的原语，它能够在同级的层之间以及从一个层到另一个交换数据。作为例子，关联于二级的层的语是“数据请求”（“Data.request”）和卡的“数据响应”（“Data.response”）以及“数据确认”（“Data.confirm”），等等的类型。

更特别地，层 13 和 23a 负责灵巧卡 2a 和主机，也就是终端 1 之间的对话。这些层能够在终端和灵巧卡 2a 的用户（未示出）之间交换信息，例如通过以“HTML”格式的超文本形式展示的菜单。它还可以布置用于发射和/或接收数据分组的合适配置。

如同上面已经指出的那样，层包括三个不同的实体。

第一层，130 或 230a 基本上由一个软件多路复用器构成。它使得在灵巧卡 2a 和主机终端 1 之间以协议数据单元的形式交换信息。它起到一个数据包交换机类似的作用。这些单元通过 2 级层（数据链接层）被发射或接收。该特定的通讯协议能够使得至少一对“智能代理”通讯。每一对的第一个代理，132，位于终端 1 这边的层 13 中，第二个，232a，位于灵巧卡 2a 这边的层 23a 中。两“智能代理”之间的连接由一个会话所关联。一个会话是一个在两代理之间的双向数据交换。

一个智能代理根据终端 1 所使用的配置可以实现全部或部分三和四级层的功能。

一个特殊的智能代理最好通过一个整数来被标识，例如 16 比特（包括在 0 到 65535 之间的数）。该标识符被在例如在构成一个目标参考或一个源参考协议数据单元中被使用。

存在有两大类的智能代理：“服务器”类型的代理，它由一个固定的标记所标识，以及“客户”类型的代理，它由一个可变的标记所标识，该标记由配置管理模块 131 或 231a 所提供。

一个会话的开启过程通常如下：一个“客户”类型的智能代理朝向一个“服务器”类型的智能代理发起会话。层 130 和 230a 管理表（未示出），它包括主机终端 1 这边和灵巧卡 2a 这边存在的智能代理的列表。

智能代理与特殊的特性和属性相关联。为说明这个概念，并且作为一个非限制性的例子，下列六个特性被关联于智能代理：

- “主机”：位于终端中的代理；
- “卡”：位于灵巧卡中的代理；
- “本地”：不与网络通讯的代理；
- “客户”：发起一个会话的代理；
- “服务器”：接收一个会话请求的代理。

智能代理允许数据交换（例如超文本的），而且启动网络交易。

配置管理模块，分别是 131 和 231a，如同已经指出的那样，类似于特殊的智能代理。例如，主机终端 1 这边的模块 131 特别是管理相对于该终端（功能模式）配置，其它存在的智能代理的列表，等等的信息。灵巧卡 2a 这边的模块 231a 具有同样的功能。该两智能代理能够一个与另一个通讯以建立一个会话。

根据一个特征，灵巧卡 2a 向主机系统，也就是说向终端 1 提交一个虚拟终端模块。为此，灵巧卡 2a 被看作是一个“WEB”服务器。

灵巧卡 2a 由导航器 10 来“寻址”。因此卡向导航器传送一个“HTML”语言的“WEB”类型页，一个“applet（小应用）”或所有其它的软件片。作为例子，“WEB”页可以具有一个欢迎页的形式，它给出了一个可能的应用选择和/或朝向外部服务器的超链接选择。

在实践中，灵巧卡 2a 最好是由一个“URL”地址（指“Universal Resource Locator（统一资源定位器）”）的使用来“寻址”，它定义了一个在终端 1 本身上的连接，而不是指向外部服务器上。作为例子，该“URL”的结构通常如下：

http://127.0.0.1:8080 (1)，

在其中 127.0.0.1 是回送的“IP”地址，而 8080 是端口号。

图 3 以简单的方式示出了系统的逻辑架构，其灵巧卡 2a 包括智能代理，其中只有两个被表示出来：一个非精确类型的智能代理，定义为 232a2 和一个被称为“WEB”类型的智能代理 232a1。逻辑栈包括符合标准 ISO 7816-3（图 2：CCa1 和 CCa2）的下部协议层，表示为 200a，“APDU”命令处理器 201a1，以及分组多路复用器 230a，该后者被与智能代理接口，特别是与“WEB”智能代理 232a1。

在终端这边，存在有两个栈，一个与因特网网络 RI 通讯，另一个与灵巧卡 2a 通讯。第一个栈包括接入网络（标准 OSI 1 和 2）的元件 11（图 2：C1 和 C2）以及表示为 100 的“TCP/IP”协议层（图 2：C3 和 C4）。该后者层与“WEB”导航器 10 接口。其它的栈包括标记为 101 并符合标准 ISO 7816-3（图 2C1 和 C2）的下部协议层，

“APDU” 指令处理器 102 和分组多路复用器 130，该后者与智能代理接口，其中只有一个被表示出来。该后者，我们假设是“网络类型”，还可以一方面通过“TCP/IP”层 100 与导航器 10 通讯，另一方面通过该相同的“TCP/IP”层和接入网络 RI 的元件 11 与因特网网络 RI 通讯。

“APDU” 指令处理器 201a 同样与一个或多个应用级的层接口，我们将简单地称应用层为应用。这些应用，如同已经指出的那样，是我们称为“cardlet（小卡）”的常规类型的应用。

综述起来，由灵巧卡 2a 提供的“WEB 服务器”功能可以由灵巧卡中的“WEB”智能代理 232a1 和终端 1 中的网络代理 132 的结合来被实现。

因此灵巧卡 2a 具有“WEB”服务器的功能。另外，根据发明方法的一个特征，不论是任何前述“CGA”类型的常规应用，A1 到 An，可以通过该“WEB”服务器，或者由存在于终端 1 中的“WEB”服务器 10，或者由位于不论是因特网网络 RI 上任意一点上的远程导航器来被激活。根据发明的方法，应用 A1 到 An 不需要被重新写入而可以如此地被应用。

根据发明的另一个特征，这些应用可以由常规类型的终端来存取，也即是说符合现有技术的。

为满足这些需求，由灵巧卡 2a 提供的“WEB”服务器功能包括了一个类似于置入在传统“WEB”服务器中的被称为“CGI”（指“Common Gateway Interface（通用网关接口）”）功能的机制。

在描述一个符合本发明的架构例子之前，该例子能够在同一灵巧卡上实现该类型功能，回顾一下“CGI”一个运行模式的主要特性是有用的。

“CGI”是一个对于“UNIX”（注册商标），“DOS”或“WIINDOWS”（注册商标）操作系统所编写的应用的来自一个“WEB”服务器的应用规范。作为例子，对于“UNIX”操作系统，规范是“CGI 1.1”，而对于“WINDOWS 95”操作系统，规范是

### “CGI 1.3”。

总是作为例子，一个下列类型的“HTTP”请求：

“http://www.host.com/cgi-bin/xxx.cgi” (2) ,

其中“host”表示一个主机系统（通常是远程的），由一个“WEB”服务器来解释，如同执行一个名为“xxx”的并且存在于该主机系统“cgi-bin”目录中的“CGI”类型的命令脚本。尽管目录名可以是任意的，通常，这是存储“CGI”类型脚本所给定的目录名。一个脚本是一系列主机系统的操作系统的指令，其最终结果被传送到前述请求的发射机“WEB”导航器。不同的语言可以用来编写该脚本，例如“PERL”（注册商标）语言。

实践上，请求通常以包括在一个“HTML”页中的表格的形式被显示在一个信息屏幕上。“HTML”语言可以将一个表格翻译成一个“URL”地址。表格包括一个或多个必需的或非必需的域，它由用户通过常规的捕获装置来填写：键盘对于文本，鼠标对于加标记的情况或被称为“无线电”的按钮，等等。表格的内容（以及可能的被称为“隐藏的”信息和指令）被发送到“WEB”服务器目的地。页的“HTML”代码描述了表格物理结构（边框，字体，色彩，以及所有其它的属性），以及要捕获的数据域的结构（名字，长度，数据类型，等等）。

传输可以根据两种主要格式类型来进行。第一种格式使用被称为“POST”的方法，第二种是被称为“GET”的方法。格式信息被表达在表格页的代码中。

然而该机制不能被直接传输到灵巧卡中，尽管它提供了符合发明特征之一的“WEB”服务器功能。

现在参照图 4 描述一个能够通过灵巧卡 2a 上的“WEB”服务器激活常规类型的任何一种应用。

在第一步骤中，一用户（未示出）从其“WEB”导航器（图 3: 10) 引用一“URL”，它可以具有下列形式：

“http://@carte:8080/xxx.html” (3)

在其中“@carte”是灵巧卡的IP地址（例如前面所描述的回送地址“127.0.0.1”：见式（1）），而“xxx.html”是相对于由灵巧卡提供的一特殊应用“xxx”的“HTML”语言页。

在第二步骤时，以前面描述的方式，灵巧卡发送一个例如表格形式的“HTML”页。

在第三步骤时，用户填写表格域并通过在“按钮”类型的特殊域上点击来向灵巧卡传送内容。

因此数据由网络代理132来发送和接收。因而数据通过分组多路复用器130（它构成了终端1这边的特定层13的部件之一），

“APDU”指令处理器102，协议层101，以传输给灵巧卡2a。然后，它通过协议层230a，“APDU”指令处理器201a，分组多路复用器230a，以由“WEB”代理230a1所接收。因此就在两智能代理之间建立一个逻辑会话，如同前面所解释的那样。

适当地要注意到定址到“WEB”代理232a1的数据以本身公知的方式，以用于“分组多路复用器”特殊应用的“APDU”指令方式来被传送。“APDU”指令处理器201a以完全类似于灵巧卡2a中“CGA”类型的其它应用，标记为A1到An，的方式选择该应用。换句话说，分组多路复用器230a被“APDU”命令处理器201a看做是一个通常的“CGA”应用。

“HTTP”请求由“WEB”代理232a1来分析，一方面它探测一个特殊目录的参考，我们在后面习惯上将其称为“cigi-smart”，另一方面探测一个特殊应用，例如在所描述例子的情况中的“xxx”。因此完整的路径表示为“cigi-smart/xxx”。

根据发明方法的一个特征，上面的实体表示一个特殊的脚本，它与一个同样特殊的应用“xxx”相关联。

在第四步骤时，脚本由一个被称为“脚本翻译器代理”的智能代理所解释，在后面我们将称其为“ATS”。该翻译可由不同的方式来实现：

a) 通过“WEB”代理232a1本身，在这种情况下它具有双重能

力；

- b) 通过唯一的脚本翻译器代理，它具有翻译存在于灵巧卡 2a 中全部脚本的能力；
- c) 通过一个我们后面将称为“ATSD”（脚本之一）的脚本代理；或者
- d) 通过一个“APDU”代理 201a 指令处理器的“APDU”代理 2010a，在该情况下它具有双能力。

“APDU”代理 2010a 是“APDU”指令处理器层 201a 的一个部件。该后者，如同已经指出的那样，是一个能够集中所有由系统发射和/或接收的“APDUU”指令，在 A1 到 An 中选择应用，而且同样提供一个智能代理类型接口的层。因此根据方法的特征之一，它能够与系统所有的智能代理通讯（通过会话），这些代理或者是位于终端 1 中或者是在灵巧卡 2a 中。

在上面 c) 的情况下，一个会话在“WEB”代理 232a1 和“ATSD”代理之一之间发起。

图 4 示出了一个翻译器代理是“ATSD”类型的架构例子。它对应于 ATS1 到 ATSn 并关联于应用 A1 到 An。被选择的应用假设是应用 Ai，会话就在“WEB”代理 232a1 和代理 ATSi 之间建立。

一个脚本翻译代理产生一个“APDU”指令序列。一个会话就在翻译代理，例如 ATSi 代理，和“APDU”代理 2010a 之间发起。因此指令就被发送向“APDU”代理 2010a。“APDU”指令处理器 201a 选择“CGA”应用 Ai（例如应用“PME”），并且向其传送翻译过的因而是常规的，可以理解的“APDU”指令。因此该应用就被正确地激活，而不必修改或重写。

应用“CGA”Ai 的响应被传送给“APDU”指令处理器 201a，“APDU”代理 2010a，然后重新到 ATSi 代理（并且更一般地到脚本翻译代理）。

根据脚本执行的成功或失败，脚本翻译器代理，例如在图 4 的例子中的代理 ATSi，产生一个“HTML”语言页，并且通过由初始请求

取得的不同层传递它，但在相反的方向上，这就被显示在可视化的屏幕 5 上（图 1A）。

在图 4 上不同的路径通过连接功能块的实线或在这些块内部的虚线来被符号化地表示。

图 5 简要地总结了刚刚描述的过程的主要步骤：

- a) 通过因特网网络 RI（或从本地终端：在两种情况下借助于一个常规导航器 10）传递一“HTTP”请求，标记为 RQ；
  - b) 灵巧卡 2a 的以表格的形式的“WEB”服务器响应，标记为 FO；
  - c) 以一个新的请求 RQ 的形式传递填好的表格；并且
  - d) 以一个“HTML”页的形式响应，标记为 PR。
- 响应另外还可以是传递一个文件，或一个软件片或“Applet”。

为实现刚刚描述的机制和功能应用，特别是“WEB”服务器功能和脚本翻译器智能代理的方法，根据一个基本特征，根据发明的方法将能够定义一个由灵巧卡保密的虚拟环境。在一个最佳实施例中，该环境与被称为多媒体类型的应用兼容。

该后一特征特别有用，因为最近的“WEB”导航器，但类型本身完全是传统的，由其性质能够构造多媒体环境（图像，动画，声音，等等。）实际上这与集成的或非集成的、能够操作多媒体文件（图像编辑器，等等）的软件工具相关联。不管怎样，导航器能够远程装载多媒体数据文件，通常是体积庞大的并且能够将其存储在硬盘，例如在终端中的，或存储在类似的大容量存储机构中。我们尤其是已经提供了能够从因特网网络的“WEB”站点实时或准实时显示视频序列或重构声音的技术。

然而，一灵巧卡，如同已经回顾的，只具有很弱的存储能力。另外，当在交换时它只能允许非常小的数据流量。因此它不能在其中记录大量的数据文件。在实践中，它不再能够设想存储多媒体数据，除非是非常短的序列，或者以特殊格式编码声音序列，例如“MIDI”编码。

在这些技术特性限制之外，还希望能够存取远程应用，并完全享受高水平的安全性，只能由灵巧卡所提供。

根据发明的方法允许该功能模式。

由灵巧卡提供安全性的多媒体虚拟环境，根据一个最佳的实施例，能够：

- 定义灵巧卡能够存取的虚拟对象；
- 提供存取该对象的方法。

图 6 简要地示出了根据发明方法的该基本观点。

一用户  $U_i$  通过终端 1 中的“WEB”导航器 10 查询灵巧卡 2a。根据一将在后面详细描述的机制，特别是由前面所描述的“WEB 服务器”功能，灵巧卡 2a 将向导航器发送能存取的、被称为虚拟的对象列表  $Obv_{p,i}$ ， $p$  是任意的索引，也就是说实际上对于它们，灵巧卡 2a 或用户  $U_i$  具有存取权。实际上，这些存取权能够被严格地连接到灵巧卡 2a 并且是不变的。它们同样可以连接到用户简档，用户  $U_i$  提供，例如，标识数据和一个口令。灵巧卡 2a 通过与存储在只读存储器中的安全数据库中的数据相比较而执行一个检验，如果比较结果是好的，就提供一个与“标识数据-口令”相关联的虚拟对象列表  $Obv_i$ 。以本身公知的方式，该第一阶段能够实现在终端和灵巧卡 2a 之间交换的数据加密的方法或实现一个安全传输协议“HTTPS”。灵巧卡同样将提供一个虚拟对象  $Obv_i$  的存取方法列表。

虚拟对象  $Obv_i$ ，如同前面指出的那样是静态的或动态的可以毫无不同地被定位在灵巧卡 2a 中，或在终端 1 中，或更一般地，在与因特网网络 RI 相连的任何系统中。根据发明的一个特征，该位置对于导航器 10 来说是“透明的”，因而对于用户  $U_i$  也是，如同将要被示出的那样。

根据发明的方法特别是要求一个后面将描述的一个虚拟文件管理系统，或“SGFV”，以及一个专用于该任务的将被称为“ATSDA/SGFV”的特殊脚本翻译器智能代理。该智能代理提供灵巧卡 2a 可以存取的虚拟对象  $Obv_i$  列表。一个特殊的“URL”地址与每

一个虚拟对象  $Obv_i$  相关联。该从“WEB”导航器 10 的“URL”调用能够通过特定于或非特定于该对象的确定调用方法来例示虚拟对象  $Obv_i$ 。

我们将首先简单地回顾一下传统的文件管理系统，此后将被称为“SGF”的主要特性。一该系统被用于存储在媒体，例如一个硬盘上的信息。信息以文件的形式被存储。一文件，是纯数据或程序指令，传统上包括一系列固定尺度的块。一非常公知的机制能够获得构成文件和它们在存储器中地址存储块的列表。

一个目录是一个特殊的文件，而其内容是文件描述符的列表。一该文件描述符包括例如下列单元：

- 文件名；
- 文件长度；
- 建立日期；
- 一个能够搜索文件块列表（第一块数，文件块数的表，等等）的参考；
- 以及指定文件特殊特性的属性（目录，读，写，执行，等等）。

第一个目录通常被称为根目录。一个不是根目录的目录被称为子目录。包含一给定文件描述符的目录是其父目录。因此“SGF”中一文件的地址是一个目录名的从根目录直到文件父目录的序列，这就定义了一个路径。作为例子，一条路径具有如下形式：

“/root/directory 1/directory 2/fine\_name” (4),

数字 1 和 2 是任意的，“root”是根目录名，而“fine\_name”是一个任意的文件名。

对于一个灵巧卡，标准 ISO 7816-4 定义了被称为“MF”（指“Master File”主文件）的根目录，被称为“DF”（指“Dedicated Files”专用文件）的子目录，和被称为“EF”（指“Elementary Files”基本文件）的基本文件。

在发明的范围内，“SGFV”文件管理系统，我们将称为虚拟，能

够定义灵巧卡 2a 能够存取的虚拟对象  $Obv_i$ 。根据发明的方法，一虚拟对象  $Obv_i$  与一个虚拟基本文件相关联。虚拟基本文件的内容由能够存取与虚拟对象  $Obv_i$  相关的信息集合而构成及在终端 1 获得一例示。

实践中，如同图 7 简要所示的那样，“SGFV”能够构成一传统“SGF”系统的子集合，并且更准确地，“SGFV”处于基本文件的内部，例如由前述标准 ISO 7816-4 所定义。

一文件描述符通常将包括下列单元：

- 文件名；
- 文件长度；
- 建立日期；
- 一能够检索文件块列表（第一块数，块数表，等等）的参考（最好是一个整数）：虚拟文件由其名或该唯一参考来标识；并且
- 指定文件特殊参考的文件属性：目录或基本文件，虚拟的或非虚拟的，直接的或间接的。

一对象由灵巧卡 2a 被例示，我们称为“直接虚拟对象”。典型地，这涉及到一个静态虚拟对象  $Obv_i$ ，它能够由导航器来操作，例如显示（图像，等等）。一虚拟对象  $Obv_i$  被由导航器 10，典型的是借助于一“applet”来例示，我们称为“间接虚拟对象”。

图 8 简要地示出了一灵巧卡系统的架构，它能够通过导航器 10 和灵巧卡 2a 例示一位于因特网网络 RI 任意地方的一虚拟对象  $Obv_i$ 。

对于前述图中公共的部件具有相同的标号，并且仅是在需要的时候才被重复描述。

图 8 上所示的架构非常类似于图 4 的架构。主要的不同是在于我们安置有一个存贮在灵巧卡 2a 中的“SGFV”8，和一个特定脚本翻译器智能代理“ATSDA/SGFV”，标记为 7。运行模式类似于图 4 所示的那样，当我们希望存取一个特殊应用  $A_i$  时。因此重新详细描写是无用的。在目前情况中，特殊应用由虚拟文件管理系统“SGFV”8 所代替。我们首先在网络智能代理 132 和“WEB”智能代理 232a1 之间建立一个会话。根据前面所解释的机制，然后在“WEB”代理 232a1

和智能代理“ATSDA/SGFV”7之间建立一个会话。

实践中，智能代理“ATSDA/SGFV”7可以通过“URL”存取，典型的类型：

“<http://www.host.com/cgi-smart/sgfv?>” (5)

其中“sgfv”是一个与智能代理“ATSDA/SGFV”7相关联的“CGI”类型的应用。上面的请求能够遍历目录树并通过一“HTML”页向导航器10“显示”它们的内容。树的“叶”是基本文件，虚拟的或非虚拟的，与超链相关联的。在“灵巧卡2a-终端1”方向上的传输以参看图4所解释的那样被实现。

换句话说，智能代理“ATSDA/SGFV”7将一“URL”地址与“SGFV”8的所有元素，目录或基本文件，关联。一目录的“URL”地址指定一个包含其元素列表的“HTML”页。一基本文件“URL”地址能够建立一个与该虚拟文件相关联的虚拟对象Obv<sub>i</sub>例示。

为说明该概念，如果我们使用上面(5)的“URL”地址，我们就获得了一个对于导航器10表示根目录内容的“HTML”页。该根目录由一组子目录和文件构成，如图9简单地所示。在该图上，我们在最上层示出了一个根目录rep#0，一个真实的基本文件fe#7，和一个真实的子目录srep#1，在中间层，以及一个虚拟的子目录rep#2和一个虚拟的基本文件fe#5，在最下层，所有二者依赖于真实的子目录srep#1，标记数字可以是任意的。

在第一阶段，智能代理“ATSDA/SGFV”7作为对所接收到的请求的响应，向导航器10传递一“HTML”页(未示出)，它以一种或另一种形式示出“SGFV”8的等级结构。页通常以例如菜单的形式被显示在可视化屏幕上(图1A:5)。每一菜单行由一个描述一子目录或一基本文件的超链接构成。更好地可以以图形的方式，并且与以描述性的文字相关联或者不相关联，显示被显示在前述屏幕上的图9的树图。还可以显示图标或复杂的形式(例如3维的)，每一个与要例示的虚拟对象相关联，并且以其自然特性(例如，一个照相机表示一个视频文件)被表示，关联或不关联一描述性的文本。

用户  $U_i$  被提示在超链接上点击（在一个节点上或在图形情况下在一分支上）。通过该动作，将获得一个所希望的虚拟对象的示例。

“SGFV” 系统 8 最好被存储在灵巧卡 2a 的可重编程类型的存储器中，例如 EEPROM 类型，如图 10 上简要所示的那样。“SGFV” 8 重新产生图 9 的树结构。

还是在所描述的例子中，在获得菜单时，菜单是在初始阶段通过在“URL”上点击来被获得的，“URL”典型地如下：

[“http://www.host.com/cgi-smart/sgfv?/file#5”](http://www.host.com/cgi-smart/sgfv?/file#5) (6)

用户  $U_i$  获得一个与图 10 上标记为 fe#5 的基本文件相关联的虚拟对象  $Obv_5$  的一个示例。同样，这将能够获得子目录的内容，参数“file#5”可以由在 (6) 中的“file#x”来代替，#x 是与子目录相关联的数字。

非虚拟的文件被记录在灵巧卡 2a 中，并且符合管理“SGF”的常用范例。它们容纳有例如密钥的数据，数据用于智能代理“ATSDA/SGFV” 7。

对于虚拟对象  $Obv_i$  的示例所必需的信息定义可以有不同的协定，例如：

- 一零长度的虚拟文件继承其父目录的存取方法；
- 一虚拟目录与一个虚拟基本文件相关联，其名称被键控（例如“虚拟”），并且它容纳有该目录的存取方法。

实际上，在可存取虚拟对象  $Obv_i$  列表之外，一智能代理“ATSDA/SGFV” 7 还必须提供对给定虚拟对象  $Obv_i$  的存取方法，这来自容纳在虚拟基本文件中的信息的全部或部分。图 11 简要地示出了该过程。

根据发明的方法，我们准备了两种方法，根据所考虑的虚拟基本文件的属性，我们将分别称其为直接的和间接的。

直接方法包括一个在过程中应用的智能代理链的描述，该过程能够存取一虚拟对象  $Obv_i$ ，并且能够在终端中获得一示例。当发起一会话时，一给定的智能代理从初始该会话的代理中接收一调用结构列

表，它在后面将被命名为“调用方法”或“Method PDU”（指“Method Protocol Data Unit（方法协议数据单元）”）。

一调用结构包括：

- 一个智能代理的标识符，通过它会话发起；
- 应用所需的数据或变量。

由前述列表定址的第一个智能代理“消费”寻址给它的第一个调用结构。接着将剩余的结构列表发送给接下来的智能代理，通过它建立一个会话，直到穷尽列表。

为说明该思想，在图 12 上简要地示出了智能代理“ATSDA/SGFV”7 和两个级联的智能代理， $232a_m$  和  $232a_n$  之间两个不同交换阶段的例子。实际上由智能代理“ATSDA/SGFV”7 所发送的调用结构的列表包括两个不同的子列表，分别在其标题上标记为#1 和#2。第一个由第一智能代理， $232a_m$  所消费，而第二个由第二智能代理  $232a_n$  所消费。一个智能代理，例如智能代理  $232a_m$ ，由一个标记或代理标识符所标识（“代理标识符#1”或“代理标识符#2”）。被定位的智能代理，并且通过它建立一个会话，由于标题（“调用结构#1”或“调用结构#2”）而获得用于它的子列表。它所获得的子列表的变量由该代理在良好运行时所使用的数据集合构成。作为例子，一段数据可以是一个文件名（非虚拟的或直接虚拟的）。

一给定的智能代理，例如智能代理  $232a_m$ ，能够在向接下来的智能代理  $232a_n$  发送之前修改剩下的调用结构列表。为此，它定位该智能代理  $232a_n$ ，并与其建立一个会话。

调用方法最好通过 ASN.1 (ISO 的“Abstract Syntax Notation 1”) 语言来被描述。

直接存取方法使得能够例示直接来自灵巧卡 2a 的一虚拟对象  $Obv_i$ 。一开始这涉及到一个静态对象。被例示的对象通常具有一个“HTML”页或一个被传输给导航器 10 的“applet”的形式。

第二个存取方法，或间接存取方法，实际上还是一个直接存取方法，但它从终端 1 中被应用，而不在灵巧卡 2a。该方法基本上被用于

例示动态的虚拟对象  $Obv_i$ 。

根据方法的该变化，作为一用于基本虚拟文件  $fe\#x$  的“URL”的响应，智能代理“ATSDA/SGFV”7 向导航器 10 传输一“HTML”页，它包括一指向与虚拟对象  $Obv_i$  相关联的直接存取方法的超链接。

两种变化可以被使用。

第一个变化在于使用“applet”。因此在存取方法上的链接是一个位于地址“@carte”的一“applet”，它本身可以由以下被指定：

- 一非虚拟文件名（也就是说一“URL”），被存储在灵巧卡 2a 上；
- 一指定一个直接虚拟文件的“URL”。

该“小应用（applet）”的一个调用参数是一个调用结构的列表，例如如同前面所指出的以 ASN.1 编码。容纳在一“HTML”页中的“小应用”被从灵巧卡 2a 或因特网网络 RI，向导航器 10 远程装载，然后由导航器强制地（硬性地）被执行。该“小应用”与第一智能代理建立一个会话，可以任意地被标记为  $232a_p$ 。与该智能代理  $232a_p$  的连接使用，例如一个“TCP/IP”客户-服务器类型的数据交换模式（也就是说被称为“JAVA 网络程序接口”的类）。“小应用”象是“TCP/IP”客户并与一个由卡的地址和端口：“@carte:port”所标识的“TCP/IP”服务器相连（该服务器同样是一个智能代理）。

图 13 简要地示出了允许由间接方法例示一个虚拟对象的交换的不同阶段。我们在该图 13 上采取前述例子的参数，在这种情况下，是虚拟基本文件  $fe\#5$ ，它由上面的配置（6）的“URL”地址所翻译。我们假设所使用的灵巧卡的地址是“@carte”而端口是：8080。请求根据前述的过程被传递到智能代理“ATSDA/SGFV”7。它向导航器发送一由“applet”构成的“HTML”页 P。为简化绘图，该“applet”的不同指令以被放置在符号`<applet>`和`</applet>`之间的“applet Code（小应用代码）”方式被总结在图 13 上。以其自身公知的方式，“applet”与一个“JAVA”类相关联，对于“虚拟终端”我们可以随意地称其为“vt.calss”。代码同样包括指示列表结构的第一智能代

理, 标记为  $232a_p$ , 的地址以及要使用的地址和端口, 表示为地址 “@carte” 和端口 8081, 的指令。要注意到, 该智能代理  $232a_p$  可以位于灵巧卡 2a 中或者在终端 1 中。

下面的阶段, 对于导航器在于通过调用结构列表向灵巧卡请求 applet, 该列表定义了 applet 的调用参数。作为响应, 卡向其传送 applet, 它将通过导航器被加载在 “Java” 虚拟机上, 在这里它将被执行。接下来的阶段, 对于导航器, 在于通过使用 “Java” 语言的 “socket” 类来调用智能代理  $232a_p$ 。

每一个智能代理, 例如  $232a_p$ , 执行一个确切的任务: 一个加密信息的解密, 口令字和/或安全数据的验证, 一文件从第一格式到其它格式的转换, 等等。尽管我们只示出了唯一的智能代理  $232a_p$ , 我们还可以预计到, 当需要的时候, 可以如同在前述的情况下 (图 12) 级联多个智能代理, 这在图 13 上以虚线被示出。如前所述, 同样每一个智能代理  $232a_p$  消费列表结构的它所使用的一部分, 并且向下面的智能代理 (未示出) 传送剩余部分, 不变或修改。

为更好地示出第一个变化, 并为了说明该思想, 我们将考虑到用户  $U_i$  希望远程装载和执行一个例如 “MP3” 格式编码的音频文件。该文件构成虚拟对象之一, 这里标记为 FS, 它由智能代理 “ATSDA/SGFV” 7 在初始化阶段所传输的 “HTML” 菜单页所提供的。图 14 简要地示出了允许例示一该虚拟对象, 标记为 FS 的步骤序列。我们假设导航器 10 没有一个与该格式相适应的阅读器。该阅读器, 标记为 LS, 将会在因特网网站上被查找, 它可以与有声音文件 FS 的网站是不同的或相同。

在所描述的例子里, 步骤序列如下:

- a) 用户  $U_i$  超链接 (文本, 图标或所有其他要查找的对象的图形化表示) 上点击: 一个请求  $I_1$  就被发送到灵巧卡 2a;
- b) 作为回应,  $R1$ , 一个 “HTML” 页通过灵巧卡 2a 被传递到终端 1 和导航器 10;
- c) 所接收到的 “HTML” 页使得导航器 10 请求一个

“applet”：询问 I2（在目前情况下，这涉及到将要寻找相应的声音阅读器 FS）；

d) 作为回应，R2，所找到的阅读器 LS 被远程装载并被安置到终端 1 中；

e) 导航器 10 重新寻址灵巧卡 2a，请求 I3，以便获得一声音文件 FS 的例示；并且

f) 作为回应，导航器 10 接收该声音文件 FS，该文件可以适于带有声音阅读器 LS 的终端 1 所读，也就是说播放。

要注意到，所有的操作对于用户  $U_i$ ，更准确地说是对于只知道灵巧卡 2a 的导航器 10，来说都是透明的。阅读器 LS（或更一般地另一“applet”）和/或被查找的虚拟对象，也就是说在例子中的文件 FS，如果它们的尺寸与灵巧卡的存储容量相匹配，将能够被存储在其中（在图 14 上以虚线表示回送）。导航器 10 不知道虚拟对象  $Obv_i$  的确切位置。只是灵巧卡 2a，更准确地说是智能代理“ATSDA/SGFV”7，知道“SGFV”8 的列表的虚拟对象的位置以及存取它的方法。

在方法的一个最佳变化中，智能代理“ATSDA/SGFV”7 还知道对于仅有给定用户  $U_i$  能够存取（授权）的虚拟对象的列表。因此这当然涉及到安全系统。术语“安全”必须在其最广泛的意义上被考虑。作为例子，这同样涉及到能够存取某些资源的通行卡，例如根据一确定的预约，或保证安全存取保密资源的卡，例如基于一授权级别。如同已经指出的那样，资源或虚拟对象  $Obv_i$  能够由交易来构成。

这些特性还被用于直接存取方法。它构成了一个根据发明的方法的特征。

根据第二个变化，在图 15 中被简要地示出，我们可以使用一个超链接，它定义了与存取方法相关的第一智能代理的“TCP/IP”地址。地址是类型：“@Agent:AgentPort”，这里“@Agent”，确切地说是所涉及到的智能代理地址，而“AgentPort”是其端口。在这种情况下，“MethodPDU”列表是一个“URL”参数。超链接将关联于，例如一个图像或一个“HTML”页 P' 的表。

因此，作为例子，一“URL”具有如下结构：

http://@Agent:AgentPort/MethodPDU?Value=xx... (7)

它能够达到一个包括如同一个“WEB TCP/IP”服务器，任意标记为 232aq，的智能代理。该智能代理 232aq 以地址“@Agent:AgentPort”来被定位，并且以参数“Value=xx...”来接收“MethodPDU”调用结构列表。

为说明该思想，我们假设虚拟对象  $Obv_i$  是一个特殊格式通过导航器 10 要显示在屏幕上的图像（图 1A: 5），并且它不具有一个合适的程序用于该显示，通常被称为观察程序或英文“viewer”。这涉及到例如一个用于终端 1 上的操作系统中的一个可执行程序，其类型为“XXX.exe”，“XXX”是该程序名。上面的在超链接（7）上的点击动作将允许重新查找该可执行程序，该程序可以位于终端 1 中或在远程系统中。

两实施变形之间的不同是，在第一种情况中，导航器被看作“强迫”请求一“applet”的装载。所有步骤是自主地实现的。在第二种情况中，用户  $U_i$  被提示在一个超链接上点击或执行一个类似的动作。

通过对前述的阅读，我们可以很容易地看到，发明可以很好地达到其确定的目的。

然而必须要清楚的是，发明并不仅限于前面所解释的实施例，特别是与图 2 到图 15 所关联的。

特别的，关于其它的脚本翻译器代理；与虚拟文件管理系统相关联的智能代理的功能可以由一个非专用代理来替换：“WEB”代理或“APDU”代理。

## 说 明 书 附 图

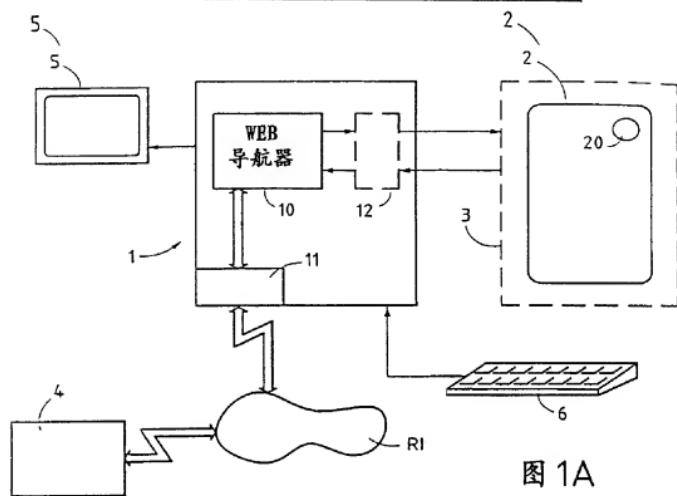


图 1A

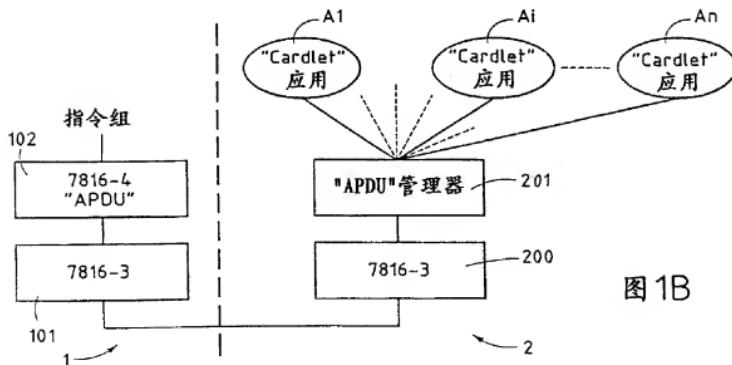


图 1B

图 2

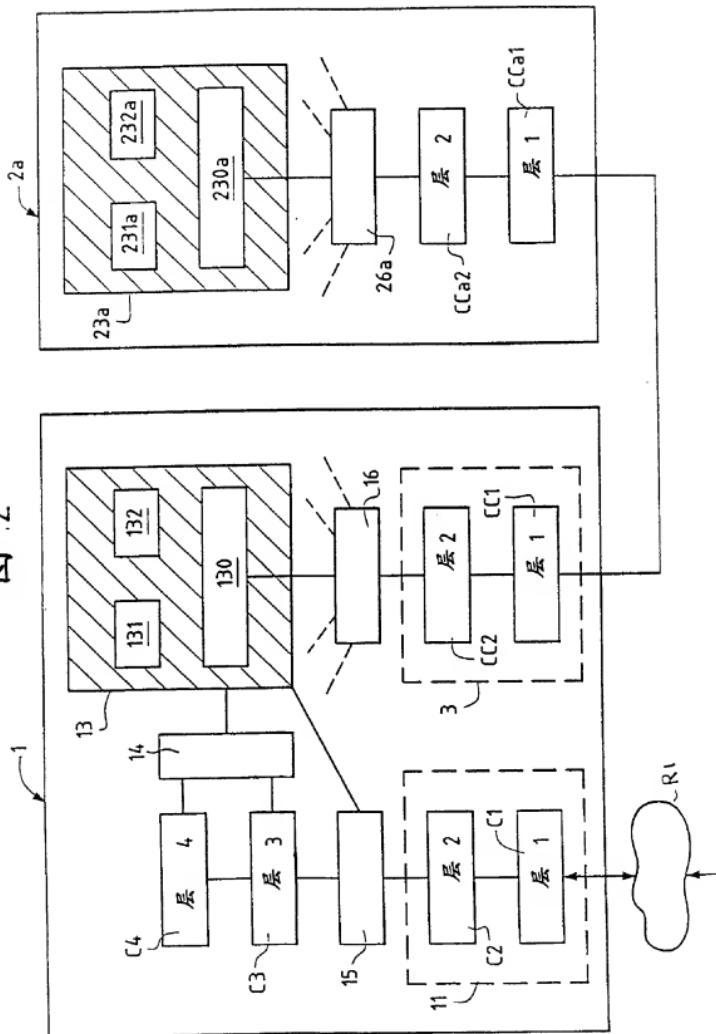


图 3

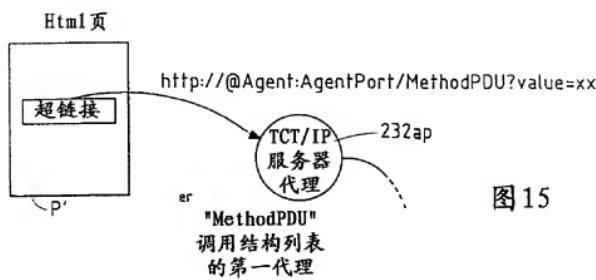
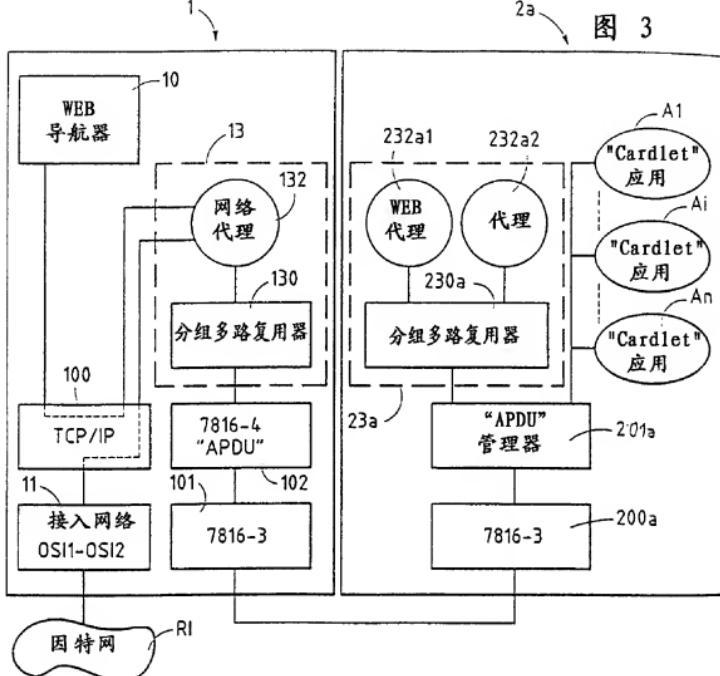


图 15

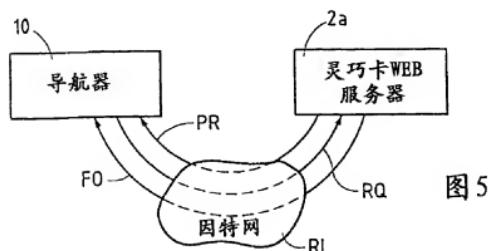
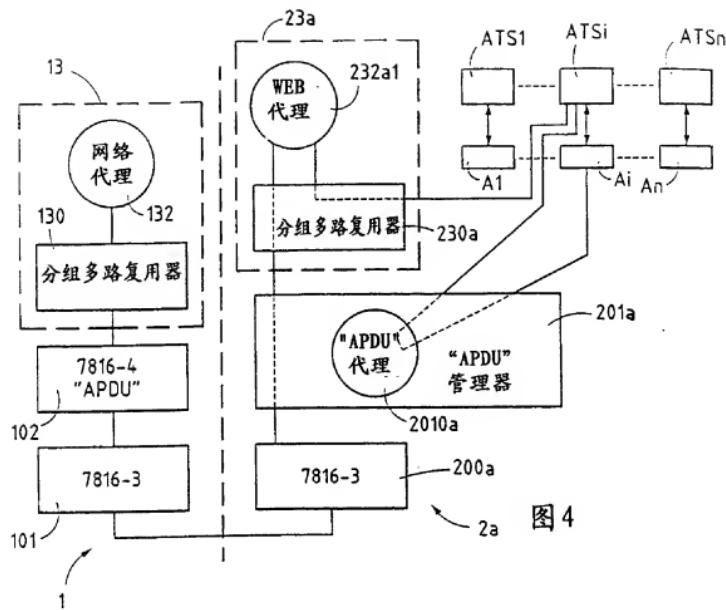


图 6

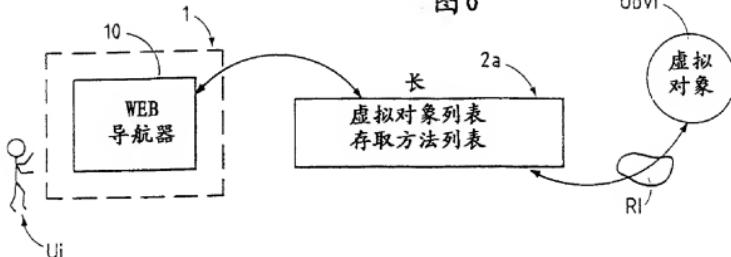


图 7

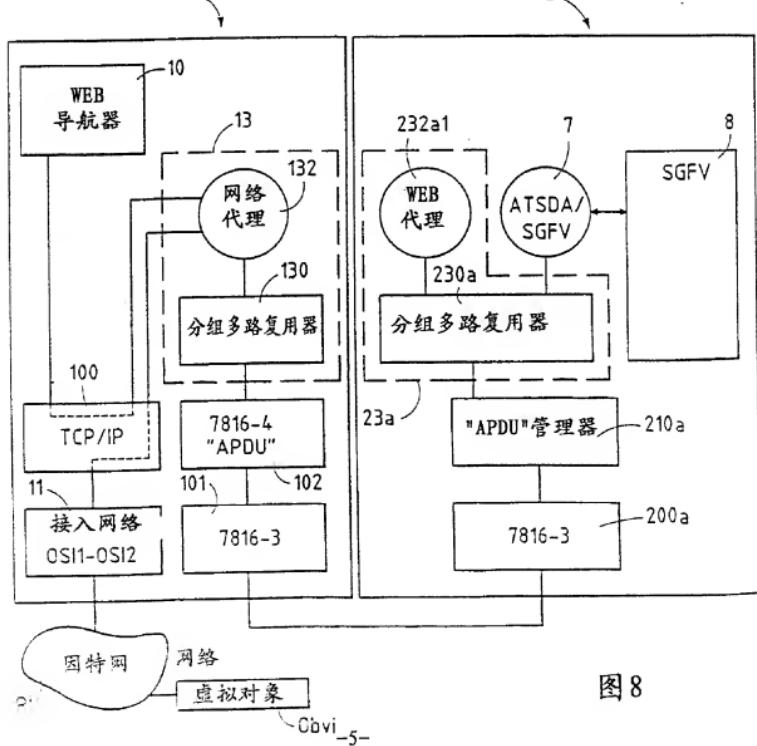


图 8

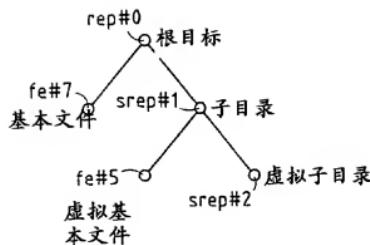


图 9

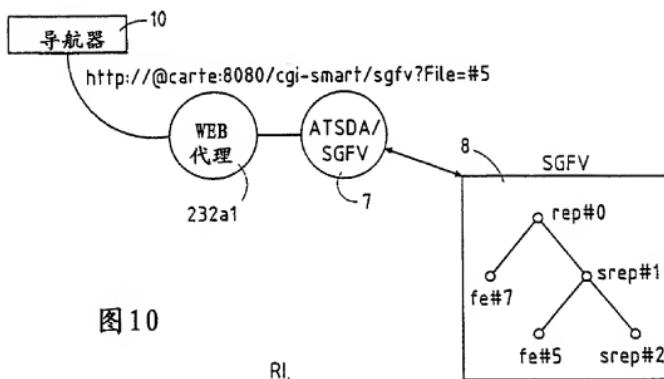


图 10

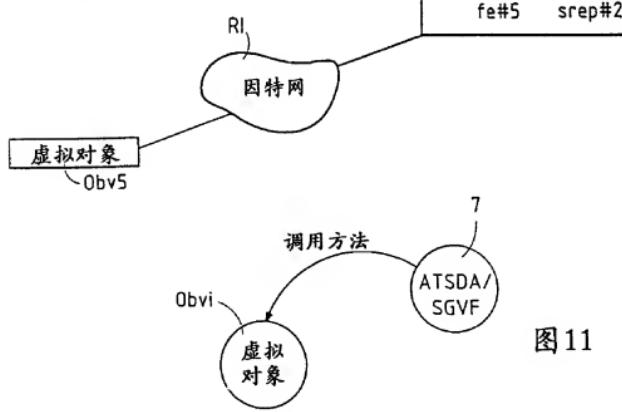


图 11

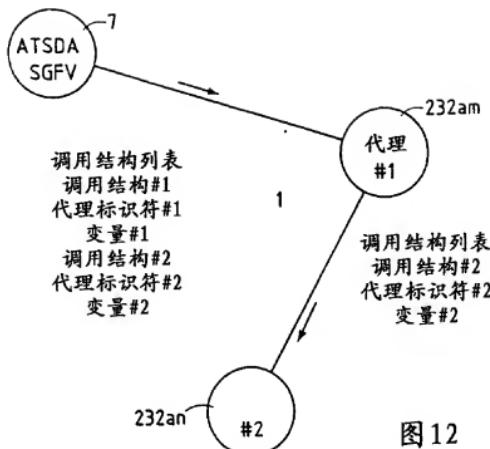


图 12

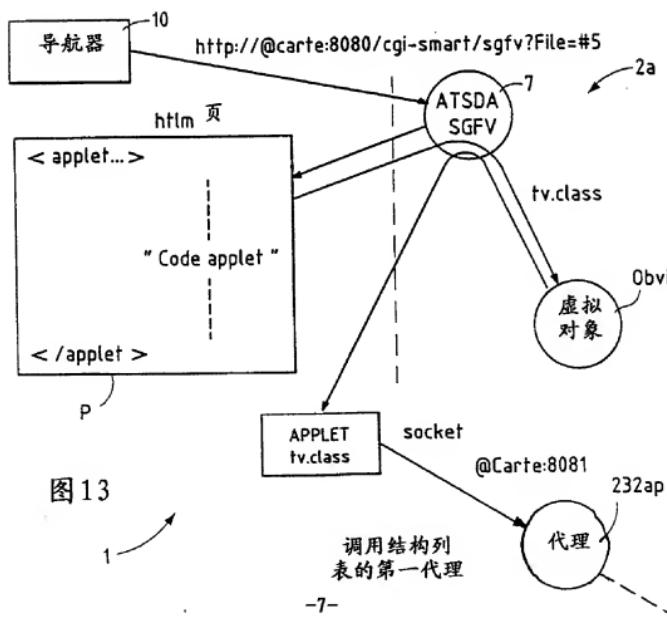


图 13

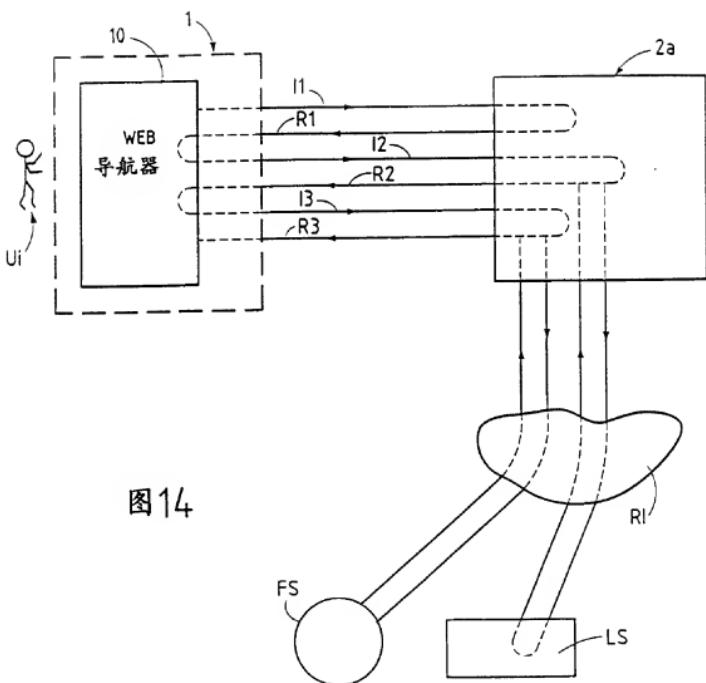


图14